

**BỘ THÔNG TIN VÀ TRUYỀN THÔNG**  
**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**  
**KHOA: CÔNG NGHỆ THÔNG TIN**

---



**BÁO CÁO THỰC HÀNH**  
**CƠ SỞ AN TOÀN THÔNG TIN**  
**BÀI THỰC HÀNH SỐ 2**

<b>Giảng viên:</b>	<b>Đinh Trường Duy</b>
<b>Nhóm môn học:</b>	<b>01</b>
<b>Tổ thực hành:</b>	<b>01</b>
<b>Sinh viên:</b>	<b>Đinh Thị Thanh Tâm</b> <b>B22DCAT253</b>

**HÀ NỘI, THÁNG 10 NĂM 2024**

## Mục lục

Labtainer sshlab: Cách tạo khóa và sử dụng ssh .....	4
Nội dung thực hiện:.....	4
Khởi động bài lab: .....	4
Kết thúc bài lab: .....	7
Labtainer nix-commands: Câu lệnh Unix/Linux .....	8
Nội dung thực hiện:.....	8
Khởi động bài lab: .....	8
Nhiệm vụ 1: Tìm hiểu các câu lệnh cơ bản .....	8
Nhiệm vụ 2: Pipe và chuyển hướng.....	14
Nhiệm vụ 3: Trợ giúp .....	15
Nhiệm vụ 4: Tìm kiếm.....	15
Nhiệm vụ 5: Kiểm soát truy cập .....	17
Nhiệm vụ 6: Quản lý tiến trình .....	19
Nhiệm vụ 7: Soạn thảo văn bản .....	20
Nhiệm vụ 8: Lịch sử .....	20
Nhiệm vụ 9: Shell script .....	21
Nhiệm vụ 10: Thực thi chương trình .....	24
Kết thúc bài lab: .....	24
Labtainer users: Giới thiệu về quản lý người dùng, nhóm và quyền đối với tệp trên hệ thống Unix .....	25
Nội dung thực hành .....	25
Khởi động bài lab: .....	25
Nhiệm vụ 1: Thêm người dùng bob.....	25
Nhiệm vụ 2: Thêm người dùng mary .....	26
Nhiệm vụ 3: Đọc tệp với bob.....	27
Nhiệm vụ 4: Leo thang đặc quyền người dùng .....	28
Kết thúc bài lab: .....	29
Labtainer acl: Danh sách điều khiển truy cập trên Linux.....	30
Nội dung thực hành .....	30
Nhiệm vụ 1: Xem lại các quyền trên các file hiện có .....	31

Nhiệm vụ 2: Cài đặt ACL trên một file .....	32
Nhiệm vụ 3: Cài đặt ACL mặc định cho 1 thư mục .....	32
Nhiệm vụ 4: Trojan Horses.....	33
Kết thúc bài lab: .....	36
KẾT LUẬN .....	37

## Labtainer sshlab: Cách tạo khóa và sử dụng ssh

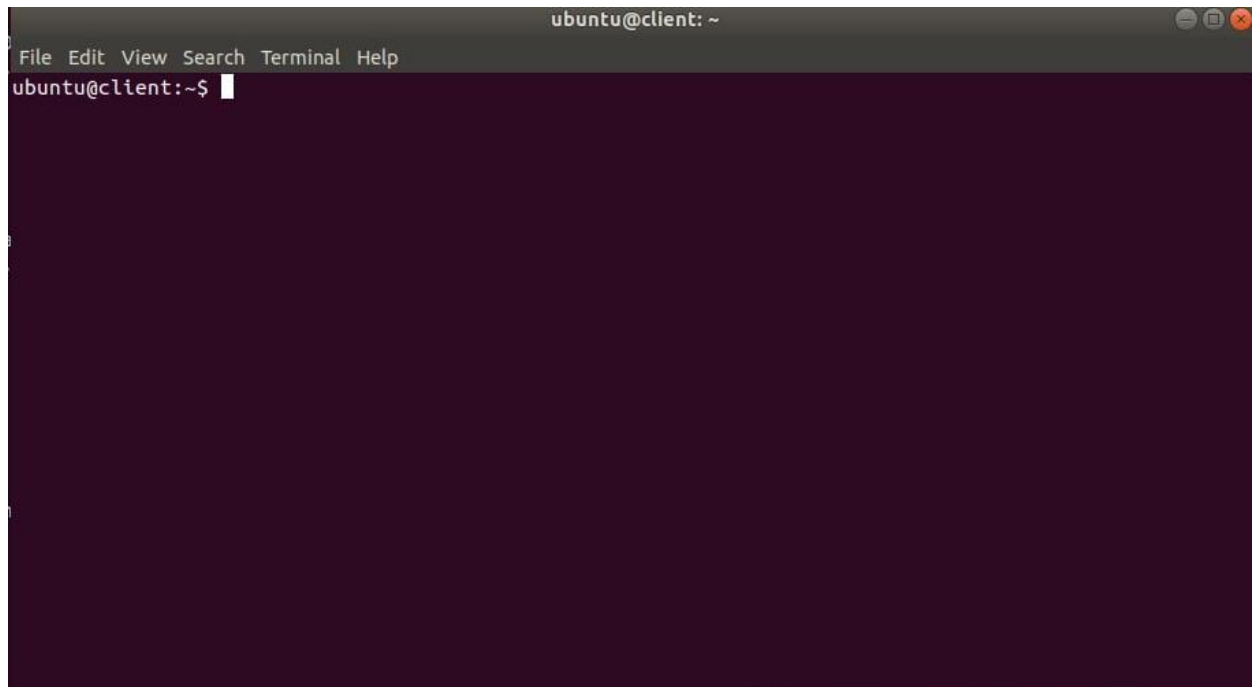
**Nội dung thực hiện:**

**Khởi động bài lab:**

Vào terminal, gõ:

*labtainer sshlab*

Sau khi khởi động xong hai terminal ảo sẽ xuất hiện, một cái là đại diện cho máy khách: **client**, một cái là đại diện cho máy chủ: **server**.



Trên terminal **client** sử dụng lệnh :

*Ifconfig*

địa chỉ IP sẽ nằm sau “inet adddr:”

```
ubuntu@client: ~  
File Edit View Search Terminal Help  
ubuntu@client:~$ ipconfig  
-su: ipconfig: command not found  
ubuntu@client:~$ ifconfig  
eth0      Link encap:Ethernet  HWaddr 02:42:ac:14:00:02  
          inet addr:172.20.0.2  Bcast:172.20.0.255  Mask:255.255.255.0  
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1  
          RX packets:56 errors:0 dropped:0 overruns:0 frame:0  
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0  
          collisions:0 txqueuelen:0  
          RX bytes:7170 (7.1 KB)  TX bytes:0 (0.0 B)  
  
lo        Link encap:Local Loopback  
          inet addr:127.0.0.1  Mask:255.0.0.0  
          UP LOOPBACK RUNNING  MTU:65536  Metric:1  
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0  
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0  
          collisions:0 txqueuelen:1000  
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)  
  
ubuntu@client:~$
```

Trên máy khách **client** sử dụng câu lệnh:

*ssh-keygen -t rsa*

để tạo cặp khóa bất đối xứng sử dụng thuật toán RSA.

```
ubuntu@client: ~  
File Edit View Search Terminal Help  
ubuntu@client:~$ ssh-keygen -t rsa  
Generating public/private rsa key pair.  
Enter file in which to save the key (/home/ubuntu/.ssh/id_rsa):  
Created directory '/home/ubuntu/.ssh'.  
Enter passphrase (empty for no passphrase):  
Enter same passphrase again:  
Your identification has been saved in /home/ubuntu/.ssh/id_rsa.  
Your public key has been saved in /home/ubuntu/.ssh/id_rsa.pub.  
The key fingerprint is:  
SHA256:sJ1YwaVuswHJHMTfpgksurs0t9hHbaf4ohUFKd1v8TU ubuntu@client  
The key's randomart image is:  
+---[RSA 2048]---+  
|    . . . . .    |  
|    = +.O        |  
|    . @ O O       |  
|    . . % O       |  
|    . =.S . . E.  |  
|    . .0=.. O ..  |  
|    + ...000 O .   |  
|    . * O+ . .    |  
|    +..+O O.      |  
+---[SHA256]-----+
```

Khóa riêng của client được lưu ở file “id\_rsa”, còn khóa công khai của client được lưu ở file “id\_rsa.pub”

Biết địa chỉ IP của server là 172.20.0.3, cần phải lưu khóa công khai lên server để server dùng nó mở tin nhắn mã hóa từ client

*ssh-copy-id -i ~/.ssh/id\_rsa.pub 172.20.0.3*

```

ubuntu@client:~$ ssh-copy-id -i ~/.ssh/id_rsa.pub 172.20.0.3
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/ubuntu/.ssh/id_rsa.pub"
The authenticity of host '172.20.0.3 (172.20.0.3)' can't be established.
ECDSA key fingerprint is SHA256:nFDnpYXdisAGpF1Zx0Bv8Xc83CDp5qYU2frYQvB7Pt8.
Are you sure you want to continue connecting (yes/no)? yes
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install all the new keys

```

Nhập mật khẩu của máy chủ là: “**ubuntu**”

```

ubuntu@172.20.0.3's password:

Number of key(s) added: 1

Now try logging into the machine, with:  "ssh '172.20.0.3'"
and check to make sure that only the key(s) you wanted were added.

ubuntu@client:~$

```

Thực hiện ssh đến máy chủ

```

cat: filetoview.txt: No such file or directory
ubuntu@client:~$ ssh ubuntu@172.20.0.3
Welcome to Ubuntu 16.04.4 LTS (GNU/Linux 4.18.0-15-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage
ubuntu@server:~$

```

Tìm và đọc tệp:

*cat filetoview.txt*

```

ubuntu@server:~$ cat filetoview.txt
# Filename: filetoview.txt
#
# Description: This is a pre-created file for each student (sshlab-server) container
#
# This file is modified when container is created
# The string below will be replaced with a keyed hash
My string is: cdddaba32dabecd491184e546ca3ce9d
ubuntu@server:~$

```

Trên terminal **server** sử dụng lệnh :

```

ubuntu@server:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 02:42:ac:14:00:03
          inet addr:172.20.0.3  Bcast:172.20.0.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:214 errors:0 dropped:0 overruns:0 frame:0
          TX packets:107 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:28606 (28.6 KB)  TX bytes:17970 (17.9 KB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

ubuntu@server:~$

```

## Kết thúc bài lab:

Trên terminal đầu tiên sử dụng câu lệnh sau để kết thúc bài lab:

*stoplab sshlab*

Trên terminal đầu tiên sử dụng câu lệnh sau để kiểm tra bài lab:

*checkwork sshlab*

```
no automated assessment for this lab
student@ubuntu:~/labtainer/labtainer-student$ checkwork sshlab
sshlab lab is not running, looking for previous results...
Labname sshlab

Student          |          sshview |  sshlogin_nopsw |
===== | ===== | ===== |
B22DCAT253      |          Y |          Y |
What is automatically assessed for this lab:

sshview: Used ssh to view the file
sshlogin_nopsw: Was able to ssh without use of a password (used ssh keys)
```

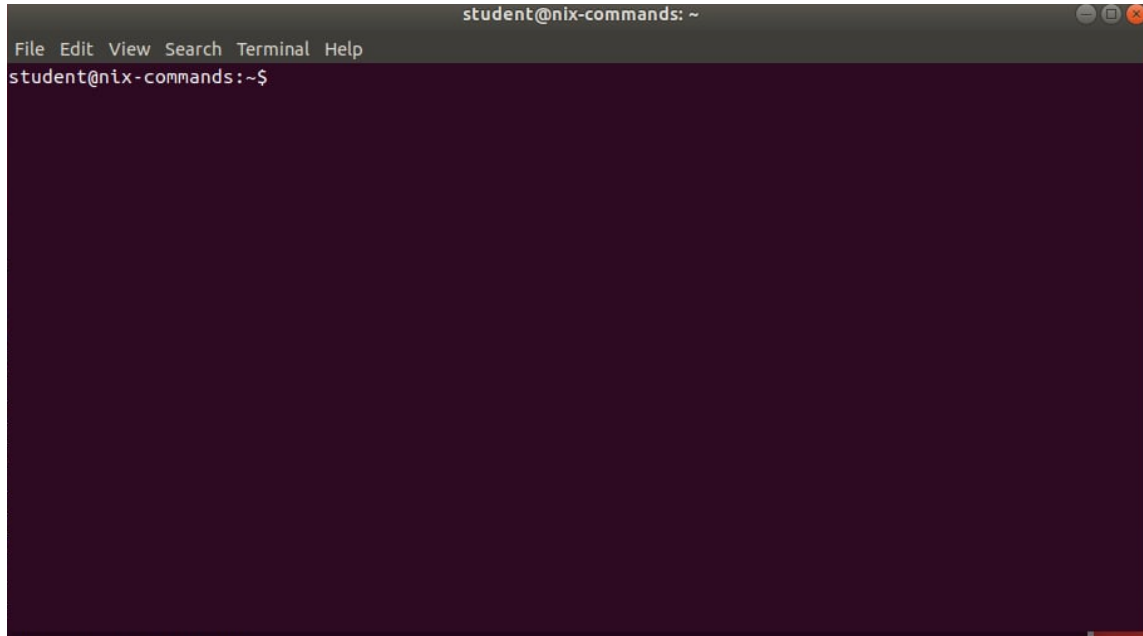
## Labtainer nix-commands: Câu lệnh Unix/Linux

**Nội dung thực hiện:**

**Khởi động bài lab:**

Vào terminal, gõ:

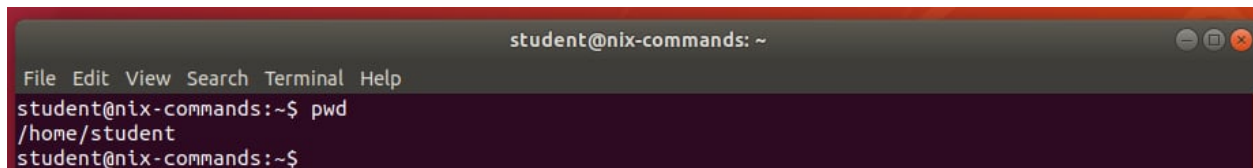
*labtainer nix-commands*



**Nhiệm vụ 1: Tìm hiểu các câu lệnh cơ bản**

Để xem thư mục gốc là gì:

*Pwd*



Liệt kê nội dung của thư mục gốc bằng cách sử dụng lệnh ls (list):

*Ls*

Lệnh *dir* tương đương với *ls*



```
student@nix-commands: ~
File Edit View Search Terminal Help
student@nix-commands:~$ pwd
/home/student
student@nix-commands:~$ ls
student@nix-commands:~$ dir
student@nix-commands:~$
```

Ngoài các tập tin bạn có thể thấy, còn có các tập tin "ẩn". Trong Unix, các tập tin ẩn là các tập tin không nhất thiết lúc nào cũng cần thấy. Bất kỳ tập tin hoặc thư mục nào bắt đầu bằng dấu '.' đều bị ẩn và được gọi là tập tin chấm (dot file). Liệt kê tất cả các đối tượng, bao gồm cả các tập tin và thư mục ẩn, bằng cách sử dụng tùy chọn -a (all), như được hiển thị dưới đây:

*ls -a*

```
student@nix-commands:~$ ls -a
.  ..  .bash_history  .bash_logout  .bashrc  .local  .profile  .sudo_as_admin_successful
student@nix-commands:~$
```

Để có cái nhìn tổng quan hơn về nội dung của thư mục hiện tại (như chủ sở hữu và kích thước), bạn cũng có thể sử dụng tùy chọn -l (dài), như được hiển thị dưới đây:

*ls -al*

```
student@nix-commands:~$ ls -al
total 40
drwxr-xr-x 1 student student 4096 Oct  6 07:01 .
drwxr-xr-x 1 root    root    4096 Jul  7 2018 ..
-rw-r--r-- 1 student student  24 Oct  6 07:04 .bash_history
-rw-r--r-- 1 student student 220 Aug 31 2015 .bash_logout
-rw-r--r-- 1 student student 3921 Oct  6 06:59 .bashrc
drwxrwxr-x 1 student student 4096 Oct  6 06:59 .local
-rw-r--r-- 1 student student 1152 Oct  6 06:59 .profile
-rw-r--r-- 1 root    root      0 Oct  6 06:59 .sudo_as_admin_successful
student@nix-commands:~$
```

Lệnh `ls -l` được sử dụng rất thường xuyên đến mức thường có một ký hiệu tắt cho nó (`ll`). Hãy thử sử dụng ký hiệu tắt cho `ls -l` như được hiển thị dưới đây:

*ll*

```

student@nix-commands:~$ ll
total 40
drwxr-xr-x 1 student student 4096 Oct  6 07:01 ./
drwxr-xr-x 1 root    root    4096 Jul  7 2018 ../
-rw-r----- 1 student student  27 Oct  6 07:05 .bash_history
-rw-r--r-- 1 student student  220 Aug 31 2015 .bash_logout
-rw-r--r-- 1 student student 3921 Oct  6 06:59 .bashrc
drwxrwxr-x 1 student student 4096 Oct  6 06:59 .local/
-rw-r--r-- 1 student student 1152 Oct  6 06:59 .profile
-rw-r--r-- 1 root    root      0 Oct  6 06:59 .sudo_as_admin_successful
student@nix-commands:~$

```

Liệt kê nội dung của thư mục bin (nơi lưu trữ nhiều lệnh cấp người dùng) bằng cách thực hiện các bước sau:

*ls /usr/bin*

```

student@nix-commands:~$ ls /usr/bin
2to3
2to3-2.7
X11
[
addpart
addr2line
apropos
apt
apt-cache
apt-cdrom
apt-confg
apt-get
apt-key
apt-mark
apt-source.sh
ar
arch
as
autoexpect
autopasswd
awk
base32
base64
basename
bashbug
bootctl
broadwayd
bsd-from
bsd-write
busctl
c++
gencat
geqn
getconf
getent
getopt
ghex
gnome-text-editor
gold
gpasswd
gpg
gpg-zip
gpgsplit
gpgv
gptic
gprof
groff
grog
grops
groovy
groups
gtbl
gtk-builder-tool
gtk-launch
gtk-update-icon-cache-3.0
h2ph
h2xs
hd
head
help2tags
hexdump
hexedit
rev
rgrep
rlogin
rlogin-cwd
rpcgen
rsh
run-mailcap
runcon
rview
rvin
savolog
scp
script
scriptreplay
sdiff
see
select-editor
sensible-browser
sensible-editor
sensible-pager
seq
set-default_gw.sh
setarch
setsid
setterm
stfp
sg
shasum
sha224sum
sha256sum
sha384sum

```

Tạo một thư mục từ bên trong thư mục gốc (tức là từ bên trong /home/student) bằng cách sử dụng lệnh mkdir:

mkdir temp

```

student@nix-commands:~$ mkdir temp

```

Để xem các thư mục

```

student@nix-commands:~$ ll
total 44
drwxr-xr-x 1 student student 4096 Oct  6 07:07 ./
drwxr-xr-x 1 root    root    4096 Jul  7 2018 ../
-rw-r----- 1 student student  53 Oct  6 07:07 .bash_history
-rw-r--r-- 1 student student  220 Aug 31 2015 .bash_logout
-rw-r--r-- 1 student student 3921 Oct  6 06:59 .bashrc
drwxrwxr-x 1 student student 4096 Oct  6 06:59 .local/
-rw-r--r-- 1 student student 1152 Oct  6 06:59 .profile
-rw-r--r-- 1 root    root      0 Oct  6 06:59 .sudo_as_admin_successful
drwxrwxr-x 2 student student 4096 Oct  6 07:07 temp/
student@nix-commands:~$

```

Thay đổi thư mục hiện tại thành thư mục mới bằng cách sử dụng lệnh cd (thay đổi thư mục):

*cd temp*

*pwd*

```
student@nix-commands:~$ cd temp
student@nix-commands:~/temp$ pwd
/home/student/temp
student@nix-commands:~/temp$
```

Liệt kê tất cả nội dung của thư mục mới này:

*ll -a*

```
student@nix-commands:~/temp$ ll -a
total 12
drwxrwxr-x 2 student student 4096 Oct  6 07:07 ./
drwxr-xr-x 1 student student 4096 Oct  6 07:07 ../
student@nix-commands:~/temp$
```

Sử dụng thư mục dotdot để liệt kê nội dung của thư mục cha:

*ll ..*

```
student@nix-commands:~/temp$ ll ..
total 44
drwxr-xr-x 1 student student 4096 Oct  6 07:07 ./
drwxr-xr-x 1 root root 4096 Jul  7 2018 ../
-rw-r--r-- 1 student student  77 Oct  6 07:10 .bash_history
-rw-r--r-- 1 student student 220 Aug 31 2015 .bash_logout
-rw-r--r-- 1 student student 3921 Oct  6 06:59 .bashrc
drwxrwxr-x 1 student student 4096 Oct  6 06:59 .local/
-rw-r--r-- 1 student student 1152 Oct  6 06:59 .profile
-rw-r--r-- 1 root root 0 Oct  6 06:59 .sudo_as_admin_successful
drwxrwxr-x 2 student student 4096 Oct  6 07:07 temp/
student@nix-commands:~/temp$
```

Sử dụng thư mục dotdot để thay đổi thư mục làm việc thành thư mục cha của thư mục temp, như được hiển thị dưới đây:

*cd ..*

*pwd*

```
student@nix-commands:~/temp$ cd ..
student@nix-commands:~$ pwd
/home/student
student@nix-commands:~$
```

Tệp hoặc thư mục có thể được di chuyển hoặc đổi tên đơn giản bằng cách sử dụng lệnh mv (move). Đổi tên thư mục temp thành temp2 bằng cách thực hiện các bước sau:

*mv temp temp2*

*ll*

```

student@nix-commands:~$ mv temp temp2
student@nix-commands:~$ ll
total 44
drwxr-xr-x 1 student student 4096 Oct  6 07:13 ./
drwxr-xr-x 1 root     root    4096 Jul  7 2018 ../
-rw----- 1 student student  104 Oct  6 07:13 .bash_history
-rw-r--r-- 1 student student  220 Aug 31 2015 .bash_logout
-rw-r--r-- 1 student student 3921 Oct  6 06:59 .bashrc
drwxrwxr-x 1 student student 4096 Oct  6 06:59 .local/
-rw-r--r-- 1 student student 1152 Oct  6 06:59 .profile
-rw-r--r-- 1 root     root     0 Oct  6 06:59 .sudo_as_admin_successful
drwxrwxr-x 2 student student 4096 Oct  6 07:07 temp2/
student@nix-commands:~$

```

Việc sao chép tệp được thực hiện bằng lệnh cp (copy). Sao chép một trong các tệp ẩn của bạn vào thư mục temp2:

```
cp .bashrc temp2
```

```
ll -a temp2
```

```

-rw-r--r-- 1 student student 3921 Oct  6 07:21 .bashrc
student@nix-commands:~$ cp .bashrc temp2/bash
student@nix-commands:~$ ll -a temp2
total 20
drwxrwxr-x 2 student student 4096 Oct  6 07:22 ./
drwxr-xr-x 1 student student 4096 Oct  6 07:14 ../
-rw-r--r-- 1 student student 3921 Oct  6 07:21 .bashrc
-rw-r--r-- 1 student student 3921 Oct  6 07:22 bash
student@nix-commands:~$

```

Hãy thử xóa thư mục temp2 bằng cách sử dụng lệnh rmdir (remove directory):

```
rmdir temp2
```

```

student@nix-commands:~$ rmdir temp2
rmdir: failed to remove 'temp2': Directory not empty

```

Nó sẽ không thành công vì vẫn còn các tệp trong thư mục đó. Xóa các tệp trong thư mục temp2 bằng cách sử dụng lệnh rm (remove):

```
rm temp2/.bas*
```

```
ll -a temp2
```

```

student@nix-commands:~$ rm temp2/.bas*
student@nix-commands:~$ ll -a temp2
total 16
drwxrwxr-x 2 student student 4096 Oct  6 07:24 ./
drwxr-xr-x 1 student student 4096 Oct  6 07:14 ../
-rw-r--r-- 1 student student 3921 Oct  6 07:22 bash
student@nix-commands:~$

```

Bây giờ xóa thư mục:

```
rmdir temp2
```

```
ll
```



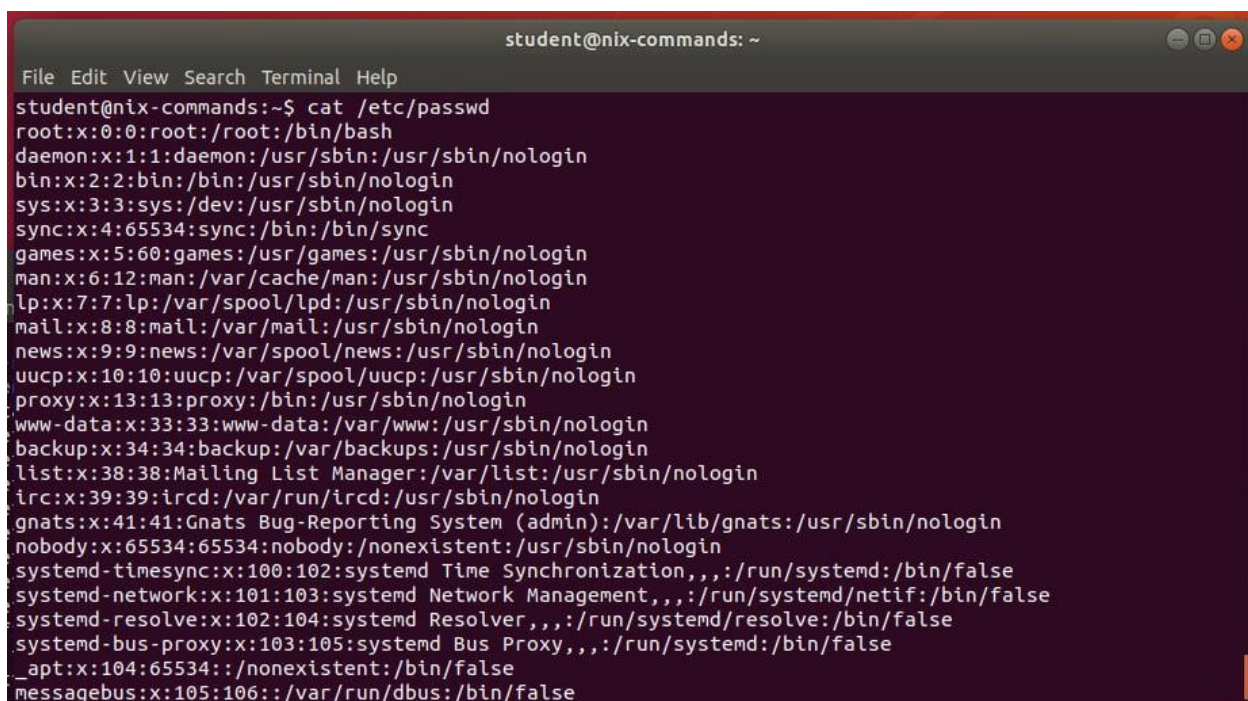
```

drwxr-xr-x 1 student student 4096 Oct  6 07:14 temp2
student@nix-commands:~$ rmdir temp2
student@nix-commands:~$ ll
total 44
drwxr-xr-x 1 student student 4096 Oct  6 07:26 ./
drwxr-xr-x 1 root     root    4096 Jul  7 2018 ../
-rw-r--r-- 1 student student  317 Oct  6 07:26 .bash_history
-rw-r--r-- 1 student student  220 Aug 31 2015 .bash_logout
-rw-r--r-- 1 student student 3921 Oct  6 06:59 .bashrc
drwxrwxr-x 1 student student 4096 Oct  6 06:59 .local/
-rw-r--r-- 1 student student 1152 Oct  6 06:59 .profile
-rw-r--r-- 1 root     root      0 Oct  6 06:59 .sudo_as_admin_successful
-rw-r--r-- 1 student student 3921 Oct  6 07:14 temp2.bash
student@nix-commands:~$

```

Để hiển thị nội dung của một tệp văn bản lên màn hình, có thể sử dụng lệnh cat (concatenate). Hiển thị nội dung của tệp password:

*cat /etc/passwd*



```

student@nix-commands: ~
File Edit View Search Terminal Help
student@nix-commands:~$ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-timesync:x:100:102:systemd Time Synchronization,,,:/run/systemd:/bin/false
systemd-network:x:101:103:systemd Network Management,,,:/run/systemd/netif:/bin/false
systemd-resolve:x:102:104:systemd Resolver,,,:/run/systemd/resolve:/bin/false
systemd-bus-proxy:x:103:105:systemd Bus Proxy,,,:/run/systemd:/bin/false
_apt:x:104:65534::/nonexistent:/bin/false
messagebus:x:105:106::/var/run/dbus:/bin/false

```

Hiển thị văn bản trên màn hình bằng cách sử dụng lệnh echo:

*echo "hello world"*

```

student@nix-commands:~$ echo "hello world"
hello world

```

Lệnh echo có thể có vẻ vô nghĩa, nhưng nó rất hữu ích, ví dụ khi viết các tập lệnh (sẽ được đề cập sau). Để xóa màn hình, ví dụ, để giảm rối, sử dụng lệnh clear:

*clear*

```
student@nix-commands: ~  
File Edit View Search Terminal Help  
student@nix-commands:~$
```

## Nhiệm vụ 2: Pipe và chuyển hướng

Chuyển hướng danh sách thư mục vào một tệp, như được hiển thị trong lệnh sau đây:

*ls /usr/bin > listing*

*ll*

```
student@nix-commands:~$ ls /usr/bin > listing  
student@nix-commands:~$ ll  
total 52  
drwxr-xr-x 1 student student 4096 Oct  6 07:32 ./  
drwxr-xr-x 1 root    root    4096 Jul  7 2018 ../  
-rw----- 1 student student  427 Oct  6 07:32 .bash_history  
-rw-r--r-- 1 student student  220 Aug 31 2015 .bash_logout  
-rw-r--r-- 1 student student 3921 Oct  6 06:59 .bashrc  
drwxrwxr-x 1 student student 4096 Oct  6 06:59 .local/  
-rw-r--r-- 1 student student 1152 Oct  6 06:59 .profile  
-rw-r--r-- 1 root    root      0 Oct  6 06:59 .sudo_as_admin_successful  
-rw-rw-r-- 1 student student 5402 Oct  6 07:32 listing  
-rw-r--r-- 1 student student 3921 Oct  6 07:14 temp2.bash  
student@nix-commands:~$
```

Có thể gắn thêm nội dung của một tệp đã tồn tại bằng cách sử dụng ">>":

*echo "testing" >> listing*

*cat listing*

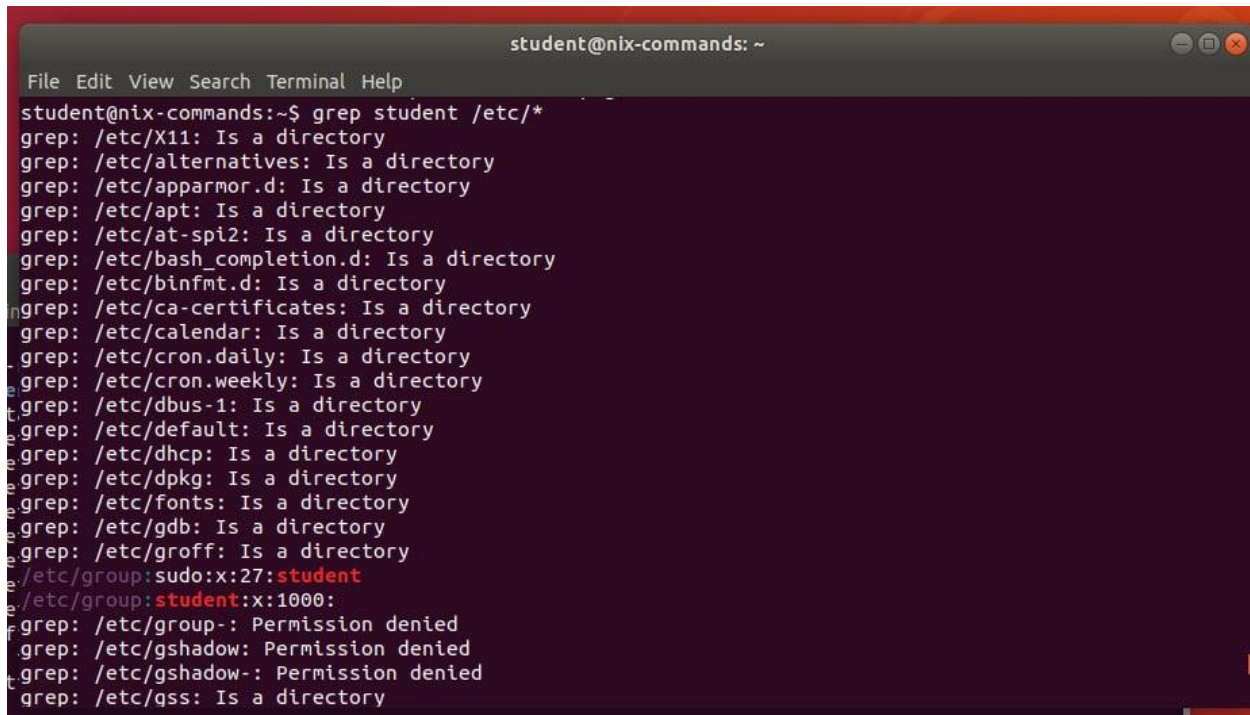
```
student@nix-commands: ~  
File Edit View Search Terminal Help  
student@nix-commands:~$ echo "testing" >> listing  
student@nix-commands:~$  
student@nix-commands:~$ cat listing  
2to3  
2to3-2.7  
X11  
[  
addpart  
addr2line  
apropos  
apt  
apt-cache  
apt-cdrom  
apt-config  
apt-get  
apt-key  
apt-mark  
apt-source.sh  
ar  
arch  
as  
autoexpect  
autopasswd  
awk  
base32
```

### Nhiệm vụ 3: Trợ giúp

### Nhiệm vụ 4: Tìm kiếm

Lệnh để tìm kiếm nội dung trong một tệp tin được gọi là grep. Tìm kiếm chuỗi "student" trong tất cả các tệp tin trong thư mục /etc:

*grep student /etc/\**



```
student@nix-commands: ~  
File Edit View Search Terminal Help  
student@nix-commands:~$ grep student /etc/*  
grep: /etc/X11: Is a directory  
grep: /etc/alternatives: Is a directory  
grep: /etc/apparmor.d: Is a directory  
grep: /etc/apt: Is a directory  
grep: /etc/at-spi2: Is a directory  
grep: /etc/bash_completion.d: Is a directory  
grep: /etc/bnfmt.d: Is a directory  
grep: /etc/ca-certificates: Is a directory  
grep: /etc/calendar: Is a directory  
grep: /etc/cron.daily: Is a directory  
grep: /etc/cron.weekly: Is a directory  
grep: /etc/dbus-1: Is a directory  
grep: /etc/default: Is a directory  
grep: /etc/dhcp: Is a directory  
grep: /etc/dpkg: Is a directory  
grep: /etc/fonts: Is a directory  
grep: /etc/gdb: Is a directory  
grep: /etc/groff: Is a directory  
e /etc/group: sudo:x:27:student  
e /etc/group: student:x:1000:  
f grep: /etc/group-: Permission denied  
t grep: /etc/gshadow: Permission denied  
t grep: /etc/gshadow-: Permission denied  
e grep: /etc/qss: Is a directory
```

Đối số đầu tiên của lệnh grep là chuỗi cần tìm kiếm (student), trong khi đối số cuối cùng là tệp tin hoặc các tệp tin cần tìm. Thư mục /etc là nơi chứa nhiều tệp tin cấu hình. Kết quả đầu ra của grep bao gồm tên tệp tin mà chuỗi được tìm thấy và đôi khi là dòng chứa chuỗi trong tệp tin. Nếu chuỗi tìm kiếm có chứa khoảng trắng, thì nó cần được đặt trong dấu ngoặc kép.

Có rất nhiều lỗi được báo cáo từ lệnh grep trước đó, vì vậy bạn có thể yêu cầu grep không báo cáo về các lỗi đó để có kết quả đầu ra sạch hơn:

*grep -s student /etc/\**



```
student@nix-commands:~$ grep -s student /etc/*  
/etc/group: sudo:x:27:student  
/etc/group: student:x:1000:  
/etc/passwd: student:x:1000:1000:/home/student:/bin/bash  
/etc/subgid: student:100000:65536  
/etc/subuid: student:100000:65536  
student@nix-commands:~$
```

Sử dụng find để tìm một tệp tin có tên "hosts" bằng cách sử dụng lệnh sau:

*find /etc -name hosts -print*

```
student@nix-commands:~$ find /etc -name hosts -print
/etc/hosts
find: '/etc/ssl/private': Permission denied
```

Có rất nhiều vấn đề về quyền truy cập khi thực thi lệnh `find` trước đó. Nhận đặc quyền root bằng cách nhập lệnh sau:

*sudo su*

```
student@nix-commands:~$ sudo su
root@nix-commands:/home/student#
root@nix-commands:/home/student#
```

Tài khoản "student" trên máy tính này là thành viên của nhóm "sudo", cho phép bạn làm việc với đặc quyền root. Tất cả các tiến trình và cửa sổ khác vẫn thực thi dành cho người dùng thông thường.

Thực thi lại lệnh `find` trước đó với đặc quyền root:

*find /etc -name hosts -print*

```
root@nix-commands:/home/student# find /etc -name hosts -print
/etc/hosts
root@nix-commands:/home/student#
```

Các ký tự đại diện có thể được sử dụng, nhưng chúng cần được đặt trong dấu ngoặc kép. Ví dụ, để tìm tất cả các tệp tin kết thúc bằng ".h", có thể sử dụng (gõ trên một dòng):

*find /usr/include -name "\*.h" -print*

```
root@nix-commands:/home/student
File Edit View Search Terminal Help
root@nix-commands:/home/student# find /usr/include -name "*.h" -print
/usr/include/regex.h
/usr/include/netinet/ether.h
/usr/include/netinet/tcp6.h
/usr/include/netinet/if_ether.h
/usr/include/netinet/ip.h
/usr/include/netinet/ip6.h
/usr/include/netinet/icmp.h
/usr/include/netinet/udp.h
/usr/include/netinet/tcp.h
/usr/include/netinet/ip_icmp.h
/usr/include/netinet/if_fddt.h
/usr/include/netinet/in.h
/usr/include/netinet/if_tr.h
/usr/include/netinet/in_sysmn.h
/usr/include/ucontext.h
/usr/include/features.h
/usr/include/execinfo.h
/usr/include/link.h
/usr/include/fmtmsg.h
/usr/include/endian.h
/usr/include/iconv.h
/usr/include/ato.h
/usr/include/dirent.h
/usr/include/rpcsvc/yppwd.h
/usr/include/rpcsvc/rstat.h
/usr/include/rpcsvc/nis_tags.h
/usr/include/rpcsvc/nis_callback.h
/usr/include/rpcsvc/rx.h
/usr/include/rpcsvc/nfs_prot.h
/usr/include/rpcsvc/yppasswd.h
/usr/include/rpcsvc/ppcint.h
/usr/include/rpcsvc/bootparam_prot.h
/usr/include/rpcsvc/rquota.h
/usr/include/rpcsvc/nln_prot.h
/usr/include/rpcsvc/key_prot.h
/usr/include/rpcsvc/nis.h
/usr/include/rpcsvc/bootparam.h
/usr/include/rpcsvc/users.h
/usr/include/rpcsvc/mount.h
/usr/include/rpcsvc/kln_prot.h
/usr/include/rpcsvc/sm_inter.h
/usr/include/rpcsvc/nislib.h
/usr/include/rpcsvc/yp.h
```



Lệnh find dưới đây được cho biết tìm kiếm trong toàn bộ cấu trúc thư mục bắt đầu từ "/usr/local"

```
find /usr/local -print
```

```
root@nix-commands:/home/student# find /usr/local -print
/usr/local
/usr/local/src
/usr/local/games
/usr/local/sbin
/usr/local/include
/usr/local/share
/usr/local/share/man
/usr/local/share/ca-certificates
/usr/local/share/fonts
/usr/local/man
/usr/local/lib
/usr/local/lib/python2.7
/usr/local/lib/python2.7/site-packages
/usr/local/lib/python2.7/dist-packages
/usr/local/lib/python2.7/dist-packages/enum-0.4.6-py2.7.egg-info
/usr/local/lib/python2.7/dist-packages/enum-0.4.6-py2.7.egg-info/SOURCES.txt
/usr/local/lib/python2.7/dist-packages/enum-0.4.6-py2.7.egg-info/top_level.txt
/usr/local/lib/python2.7/dist-packages/enum-0.4.6-py2.7.egg-info/installed-files.txt
/usr/local/lib/python2.7/dist-packages/enum-0.4.6-py2.7.egg-info/PKG-INFO
/usr/local/lib/python2.7/dist-packages/enum-0.4.6-py2.7.egg-info/dependency_links.txt
/usr/local/lib/python2.7/dist-packages/enum-0.4.6-py2.7.egg-info/requires.txt
/usr/local/lib/python2.7/dist-packages/enum-0.4.6-py2.7.egg-info/not-zip-safe
/usr/local/lib/python2.7/dist-packages/enum.pyc
/usr/local/lib/python2.7/dist-packages/enum.py
/usr/local/lib/python2.7/dist-packages/notify_simple-1.1.7-py2.7.egg-info
/usr/local/lib/python2.7/dist-packages/notify_simple-1.1.7-py2.7.egg-info/SOURCES.txt
/usr/local/lib/python2.7/dist-packages/notify_simple-1.1.7-py2.7.egg-info/top_level.txt
/usr/local/lib/python2.7/dist-packages/notify_simple-1.1.7-py2.7.egg-info/installed-files.txt
/usr/local/lib/python2.7/dist-packages/notify_simple-1.1.7-py2.7.egg-info/PKG-INFO
/usr/local/lib/python2.7/dist-packages/notify_simple-1.1.7-py2.7.egg-info/dependency_links.txt
/usr/local/lib/python2.7/dist-packages/notify_simple/notify_simple.pyc
/usr/local/lib/python2.7/dist-packages/notify_simple/_int_.py
/usr/local/lib/python2.7/dist-packages/notify_simple/notify_simple.py
/usr/local/lib/python2.7/dist-packages/notify_simple/_int_.pyc
/usr/local/lib/python2.7/dist-packages/notify_simple/_version_.pyc
/usr/local/lib/python2.7/dist-packages/parse.py
/usr/local/lib/python2.7/dist-packages/parse.pyc
/usr/local/lib/python2.7/dist-packages/parse-1.8.4-py2.7.egg-info
/usr/local/lib/python2.7/dist-packages/parse-1.8.4-py2.7.egg-info/SOURCES.txt
```

### Quay về quyền truy cập của người dùng thông thường:

*exit*

```
root@nix-commands:/home/student# exit
exit
student@nix-commands:~$
```

## Nhiệm vụ 5: Kiểm soát truy cập

Hiện thị nội dung của thư mục home:

 $cd$ 

*ll -a*

```

exit
student@nix-commands:~$ cd
student@nix-commands:~$ ll -la
total 52
drwxr-xr-x 1 student student 4096 Oct 6 07:32 ./
drwxr-xr-x 1 root root 4096 Jul 7 2018 ../
-rw----- 1 student student 582 Oct 6 10:48 .bash_history
-rw-r--r-- 1 student student 220 Aug 31 2015 .bash_logout
-rw-r--r-- 1 student student 3921 Oct 6 06:59 .bashrc
drwxrwxr-x 1 student student 4096 Oct 6 06:59 .local/
-rw-r--r-- 1 student student 1152 Oct 6 06:59 .profile
-rw-r--r-- 1 root root 0 Oct 6 06:59 .sudo_as_admin_successful
-rw-rw-r-- 1 student student 5410 Oct 6 07:33 listing
-rw-r--r-- 1 student student 3921 Oct 6 07:14 temp2.bash
student@nix-commands:~$

```

Quyền truy cập được hiển thị ở phía bên trái của đầu ra của mỗi đối tượng. Quyền truy cập được phân chia thành người dùng (tức chủ sở hữu), nhóm và những người khác. Mỗi

phần được chia thành ba quyền truy cập: đọc (r), ghi (w) và thực thi (x). Nếu quyền truy cập đã được cấp, thì nó được hiển thị, nếu không, dấu '-' cho biết quyền truy cập không được cấp. Trong ví dụ trên, người dùng (hoặc chủ sở hữu tập tin) có tất cả ba quyền truy cập, trong khi nhóm chỉ có quyền đọc và ghi, và những người khác chỉ có quyền đọc. Dấu '-' ở đầu cho biết đó là một tập tin, trong khi 'd' cho biết đó là một thư mục.

Để thay đổi quyền truy cập trên một tập tin hoặc thư mục, ta sử dụng lệnh `chmod` (change mode). Có nhiều cách để sử dụng giao diện này, nhưng ví dụ này sẽ tập trung vào những cách dễ nhớ và hiệu nhất. (Giao diện đồ họa là dễ nhất).

Để chỉ định rằng quyền truy cập của người dùng cần được thay đổi, sử dụng 'u'. Đối với nhóm, sử dụng 'g', và đối với những người khác, sử dụng 'o'. Để đọc, ghi và thực thi, sử dụng 'r', 'w' và 'x' tương ứng. Ví dụ, thay đổi quyền truy cập trên tập tin `.bashrc` để mọi người có thể ghi vào nó:

```
ll .bashrc
```

```
student@nix-commands:~$ ll .bashrc
-rw-r--r-- 1 student student 3921 Oct  6 06:59 .bashrc
```

```
chmod o+w .bashrc
```

```
student@nix-commands:~$ chmod o+w .bashrc
```

```
ll .bashrc
```

```
student@nix-commands:~$ ll .bashrc
-rw-r--rw- 1 student student 3921 Oct  6 06:59 .bashrc
student@nix-commands:~$
```

"o+w" có nghĩa là "Thêm quyền ghi cho những người khác". Bây giờ hãy gỡ bỏ quyền này:

```
chmod o-w .bashrc
```

```
ll .bashrc
```

```
student@nix-commands:~$ chmod o-w .bashrc
student@nix-commands:~$
student@nix-commands:~$ ll .bashrc
-rw-r--r-- 1 student student 3921 Oct  6 06:59 .bashrc
student@nix-commands:~$
```

Có thể thực hiện nhiều thay đổi cùng một lúc. Thực hiện các thay đổi sau đây để cấp quyền đọc, ghi và thực thi cho người dùng, nhóm và những người khác trên tập tin `.bashrc`:

```
chmod ugo+rw .bashrc
```

```
ll .bashrc
```

```
student@nix-commands:~$ chmod ugo+rw .bashrc
student@nix-commands:~$ ll .bashrc
-rwxrwxrwx 1 student student 3921 Oct  6 06:59 .bashrc*
student@nix-commands:~$
```

trong đó, người dùng, nhóm và những người khác đều được cấp quyền đọc, ghi và thực thi cho tập tin.

Để thay đổi quyền truy cập sao cho nhóm và những người khác chỉ có quyền đọc, thực hiện các bước sau đây:

*chmod go=r .bashrc*

*ll .bashrc*

```
student@nix-commands:~$ chmod go=r .bashrc
student@nix-commands:~$ ll .bashrc
-rwxr--r-- 1 student student 3921 Oct  6 06:59 .bashrc*
student@nix-commands:~$
```

## Nhiệm vụ 6: Quản lý tiến trình

Để hiển thị danh sách các tiến trình đang thực thi, ta sử dụng lệnh ps (process status). Nhập lệnh sau đây mà không có bất kỳ đối số nào:

*ps*

```
student@nix-commands:~$ ps
  PID TTY          TIME CMD
   212 pts/1        00:00:01 bash
  1429 pts/1        00:00:00 ps
```

Để hiển thị tất cả các tiến trình đang chạy (kể cả những tiến trình không liên quan đến terminal ), nhập lệnh sau đây:

*ps ax*

```
student@nix-commands:~$ ps ax
  PID TTY          STAT TIME  COMMAND
    1 ?           Ss   0:00 /sbin/init --log-target=journal
   22 ?           Ss   0:01 /lib/systemd/systemd-journald
   33 ?           Ss   0:00 /usr/bin/dbus-daemon --system --address=systemd: --nofork --nopidfile --
  138 ?          Ssl   0:00 /usr/sbin/rsyslogd -n
  199 pts/1        Ss   0:00 bash -l -c bash
  211 pts/1        S    0:00 su - student
  212 pts/1        S    0:01 -su
 1446 pts/1      R+   0:00 ps ax
student@nix-commands:~$
```

Nếu một tiến trình bị treo và không thể kết thúc, PID được hiển thị cho tiến trình đó có thể được sử dụng để chấm dứt nó. Ví dụ, nếu một tiến trình có PID là 11076 cần được chấm dứt, sử dụng lệnh sau:

*kill -9 11076*

```
student@nix-commands:~$ kill -9 138
-su: kill: (138) - Operation not permitted
```

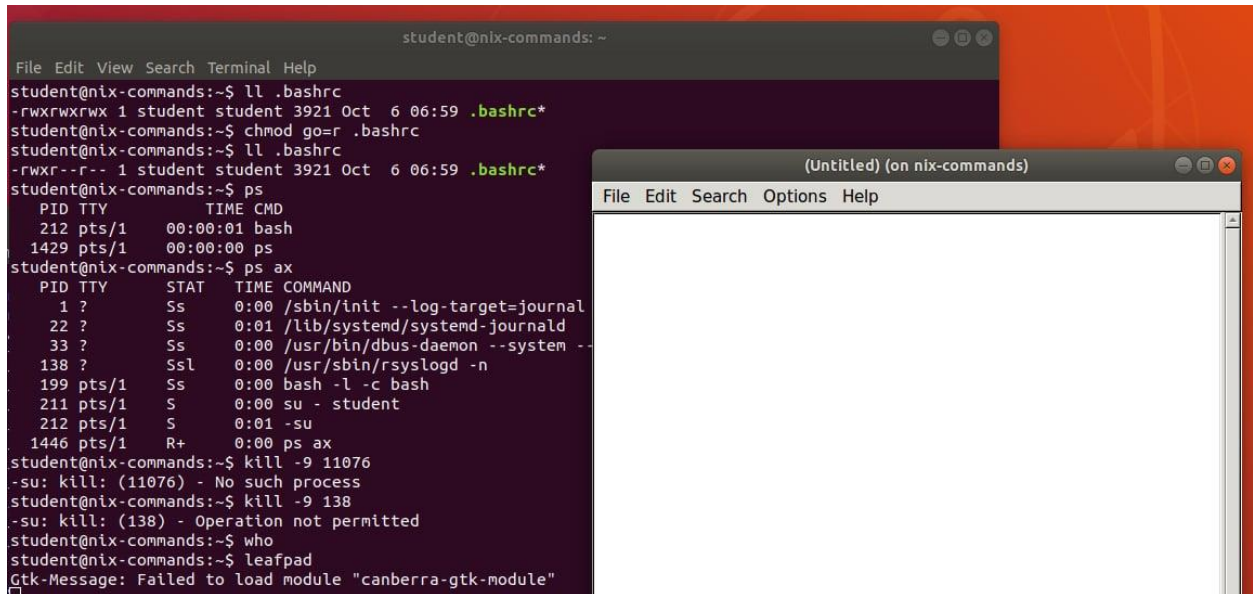
Để xem ai đang đăng nhập vào hệ thống hiện tại, ta sử dụng lệnh sau:

*who*

```
student@nix-commands:~$ who
student@nix-commands:~$
```

## Nhiệm vụ 7: Soạn thảo văn bản

Có thể gọi trình soạn thảo leafpad bằng cách gõ lệnh leafpad tại dấu nhắc lệnh.



## Nhiệm vụ 8: Lịch sử

Mặc định, hầu hết các shell Unix theo dõi các lệnh bạn đã nhập. Nhập lệnh sau để xem các lệnh đã nhập với người dùng student:

*history*

```
student@nix-commands: ~  
File Edit View Search Terminal Help  
student@nix-commands:~$ history  
1  pwd  
2  ls  
3  dir  
4  ls -a  
5  ls -al  
6  ll  
7  ls /usr/bin  
8  mkdir temp  
9  ll  
10 cd temp  
11 pwd  
12 ll -a  
13 ll ..  
14 cd ..  
15 pwd  
16 mv temp temp2  
17 ll  
18 cp .bashrc temp2.bash  
19 ll -2 temp2  
20 ll -a temp2  
21 ll -a temp2  
22 cp .bashrc temp2  
23 ll -a temp2  
24 cp .bashrc temp2/bash
```

Một cách nhanh hơn để chạy lệnh grep cuối cùng là sử dụng ký tự '!' như được hiển thị dưới đây:

*!grep*

```
student@nix-commands:~$ !grep  
grep -s student /etc/*  
/etc/group:sudo:x:27:student  
/etc/group:student:x:1000:  
/etc/passwd:student:x:1000:1000::/home/student:/bin/bash  
/etc/subgid:student:100000:65536  
/etc/subuid:student:100000:65536  
student@nix-commands:~$
```

## Nhiệm vụ 9: Shell script

Ping thường được sử dụng để xác định xem một hệ thống từ xa có phản hồi cho hoạt động mạng cấp thấp hay không. Thực hiện lệnh ping sau đây:

*ping google.com*



```
student@nix-commands: ~
File Edit View Search Terminal Help
student@nix-commands:~$ ping google.com
PING google.com (142.250.197.110) 56(84) bytes of data.
64 bytes from nchkg-a-i-f14.1e100.net (142.250.197.110): icmp_seq=1 ttl=127 time=27.5 ms
64 bytes from nchkg-a-i-f14.1e100.net (142.250.197.110): icmp_seq=2 ttl=127 time=46.0 ms
64 bytes from nchkg-a-i-f14.1e100.net (142.250.197.110): icmp_seq=3 ttl=127 time=37.0 ms
64 bytes from nchkg-a-i-f14.1e100.net (142.250.197.110): icmp_seq=4 ttl=127 time=70.9 ms
64 bytes from nchkg-a-i-f14.1e100.net (142.250.197.110): icmp_seq=5 ttl=127 time=27.8 ms
64 bytes from nchkg-a-i-f14.1e100.net (142.250.197.110): icmp_seq=6 ttl=127 time=25.2 ms
64 bytes from nchkg-a-i-f14.1e100.net (142.250.197.110): icmp_seq=7 ttl=127 time=25.1 ms
64 bytes from nchkg-a-i-f14.1e100.net (142.250.197.110): icmp_seq=8 ttl=127 time=27.7 ms
64 bytes from nchkg-a-i-f14.1e100.net (142.250.197.110): icmp_seq=9 ttl=127 time=32.2 ms
64 bytes from nchkg-a-i-f14.1e100.net (142.250.197.110): icmp_seq=10 ttl=127 time=27.1 ms
64 bytes from nchkg-a-i-f14.1e100.net (142.250.197.110): icmp_seq=11 ttl=127 time=25.7 ms
64 bytes from nchkg-a-i-f14.1e100.net (142.250.197.110): icmp_seq=12 ttl=127 time=28.1 ms
64 bytes from nchkg-a-i-f14.1e100.net (142.250.197.110): icmp_seq=13 ttl=127 time=30.1 ms
64 bytes from nchkg-a-i-f14.1e100.net (142.250.197.110): icmp_seq=14 ttl=127 time=28.0 ms
64 bytes from nchkg-a-i-f14.1e100.net (142.250.197.110): icmp_seq=15 ttl=127 time=36.7 ms
64 bytes from nchkg-a-i-f14.1e100.net (142.250.197.110): icmp_seq=16 ttl=127 time=81.8 ms
64 bytes from nchkg-a-i-f14.1e100.net (142.250.197.110): icmp_seq=17 ttl=127 time=28.3 ms
64 bytes from nchkg-a-i-f14.1e100.net (142.250.197.110): icmp_seq=18 ttl=127 time=26.8 ms
64 bytes from nchkg-a-i-f14.1e100.net (142.250.197.110): icmp_seq=19 ttl=127 time=35.7 ms
64 bytes from nchkg-a-i-f14.1e100.net (142.250.197.110): icmp_seq=20 ttl=127 time=28.1 ms
64 bytes from nchkg-a-i-f14.1e100.net (142.250.197.110): icmp_seq=21 ttl=127 time=36.5 ms
64 bytes from nchkg-a-i-f14.1e100.net (142.250.197.110): icmp_seq=22 ttl=127 time=40.3 ms
```

Lệnh ping trên xác định xem máy tính với địa chỉ IP đã cho có hoạt động hay không. Khi bạn đã xem đủ, nhấn Ctrl-C để kết thúc ping. Hãy giả sử một ví dụ đơn giản: mỗi sáng, việc đầu tiên bạn làm là kiểm tra xem các máy chủ quan trọng nhất của bạn có thể truy cập được hay không. Có thể thực hiện các lệnh sau (nhưng không bây giờ):

*ping mail*

*ping payroll*

*ping printer*

*ping router*

Thay đổi các lệnh ping để phù hợp với một script. Nhập các lệnh sau:

*ping -c 1 -w 1 google.com*

```
student@nix-commands:~$ ping -c 1 -w 1 google.com
ping: bad number of packets to transmit.
student@nix-commands:~$
```

Thay vì ping liên tục cho đến khi bị ngắt bởi người dùng, lệnh trên chỉ sẽ ping một lần (-c 1) và chỉ chờ một giây (-w 1) để nhận phản hồi. Điều đó cải thiện, nhưng không thực sự muốn nhìn thấy tất cả các đầu ra đó thử lệnh:

*ping -c 1 -w 1 google.com > /dev/null*

```
student@nix-commands:~$ ping -c 1 -w 1 google.com > /dev/null
ping: bad number of packets to transmit.
```

Bạn đã chuyển hướng tất cả đầu ra vào một lỗ đen mà không có gì trả về. Như vậy, đã loại bỏ tất cả đầu ra, nhưng bây giờ bạn không biết liệu ping đã thành công hay không. Vì vậy, thêm thứ sau (tất cả trên cùng một dòng lệnh):

```
ping -c 1 -w 1 google.com > /dev/null && echo Up
```

```
tstudent@nix-commands:~$ ping -c 1 -w 1 google.com > /dev/null && echo up
ping: bad number of packets to transmit.
```

Nếu ping thành công, lệnh sau "&&" được thực thi, ngược lại thì không được thực thi. Trong trường hợp này, "Up" được hiển thị trên màn hình nếu ping thành công. Bây giờ thực hiện viết một script. Khởi chạy một trình soạn thảo, ví dụ như leafpad, và nhập các dòng sau đây:

```
echo
```

```
echo "Trying Google"
```

```
ping -c 1 -w 1 google.com > /dev/null && echo Up
```

```
echo
```

```
echo "Trying Bing"
```

```
ping -c 1 -w 1 bing.com > /dev/null && echo Up
```

```
echo
```

```
echo "Trying NPS"
```

```
ping -c 1 -w 1 nps.edu > /dev/null && echo Up
```

Lưu tệp tin trong thư mục home với tên là "pinger", sau đó thoát khỏi trình soạn thảo.

Làm cho tệp tin có thể thực thi và thử nghiệm bằng cách thực hiện các bước sau:

```
ll
```

```
student@nix-commands:~$ ll
total 64
drwxr-xr-x 1 student student 4096 Oct 6 11:13 ./
drwxr-xr-x 1 root root 4096 Jul 7 2018 ../
-rw----- 1 student student 999 Oct 6 11:13 .bash_history
-rw-r--r-- 1 student student 220 Aug 31 2015 .bash_logout
-rwxr--r-- 1 student student 3921 Oct 6 06:59 .bashrc*
drwx----- 4 student student 4096 Oct 6 11:04 .config/
drwxrwxr-x 1 student student 4096 Oct 6 06:59 .local/
-rw-r--r-- 1 student student 1152 Oct 6 06:59 .profile
-rw-r--r-- 1 root root 0 Oct 6 06:59 .sudo_as_admin_successful
-rw-rw-r-- 1 student student 5410 Oct 6 07:33 listing
-rw-rw-r-- 1 student student 222 Oct 6 11:13 pinger
-rw-r--r-- 1 student student 3921 Oct 6 07:14 temp2.bash
-rw-rw-r-- 1 student student 9 Oct 6 11:04 test
student@nix-commands:~$
```

```
chmod u+x pinger
```

*ll*

*./pinger*

```
student@nix-commands: ~
File Edit View Search Terminal Help
student@nix-commands:~$ chmod u+x pinger
student@nix-commands:~$ ll
total 64
drwxr-xr-x 1 student student 4096 Oct  6 11:13 ./
drwxr-xr-x 1 root    root    4096 Jul  7 2018 ../
-rw-r--r-- 1 student student 1019 Oct  6 11:13 .bash_history
-rw-r--r-- 1 student student  220 Aug 31 2015 .bash_logout
-rwxr--r-- 1 student student 3921 Oct  6 06:59 .bashrc*
drwx----- 4 student student 4096 Oct  6 11:04 .config/
drwxrwxr-x 1 student student 4096 Oct  6 06:59 .local/
-rw-r--r-- 1 student student 1152 Oct  6 06:59 .profile
-rw-r--r-- 1 root    root      0 Oct  6 06:59 .sudo_as_admin_successful
-rw-rw-r-- 1 student student 5410 Oct  6 07:33 listing
-rwxrw-r-- 1 student student  222 Oct  6 11:13 pinger*
-rw-r--r-- 1 student student 3921 Oct  6 07:14 temp2.bash
-rw-rw-r-- 1 student student   9 Oct  6 11:04 test
student@nix-commands:~$ ./pinger

Trying Google
Up

Trying Bing
Up

Trying NPS
```

## Nhiệm vụ 10: Thực thi chương trình

Có thể xác định đường dẫn đến tệp tin mà shell sẽ thực thi bằng cách sử dụng lệnh *which*:

*which ls*

```
Trying NPS
student@nix-commands:~$ which ls
/bin/ls
```

## Kết thúc bài lab:

Trên terminal đầu tiên sử dụng câu lệnh sau để kết thúc bài lab:

*Stoplab nix-commands*

Trên terminal đầu tiên sử dụng câu lệnh sau để kiểm tra bài lab:

*Checkwork nix-commands*

```
student@ubuntu:~/labtainer/labtainer-student$ checkwork nix-commands
nix-commands lab is not running, looking for previous results...
Labname nix-commands

Student
=====
B22DCAT253
What is automatically assessed for this lab:

No automated assessment for this lab
student@ubuntu:~/labtainer/labtainer-student$
```



## Labtainer users: Giới thiệu về quản lý người dùng, nhóm và quyền đối với tệp trên hệ thống Unix

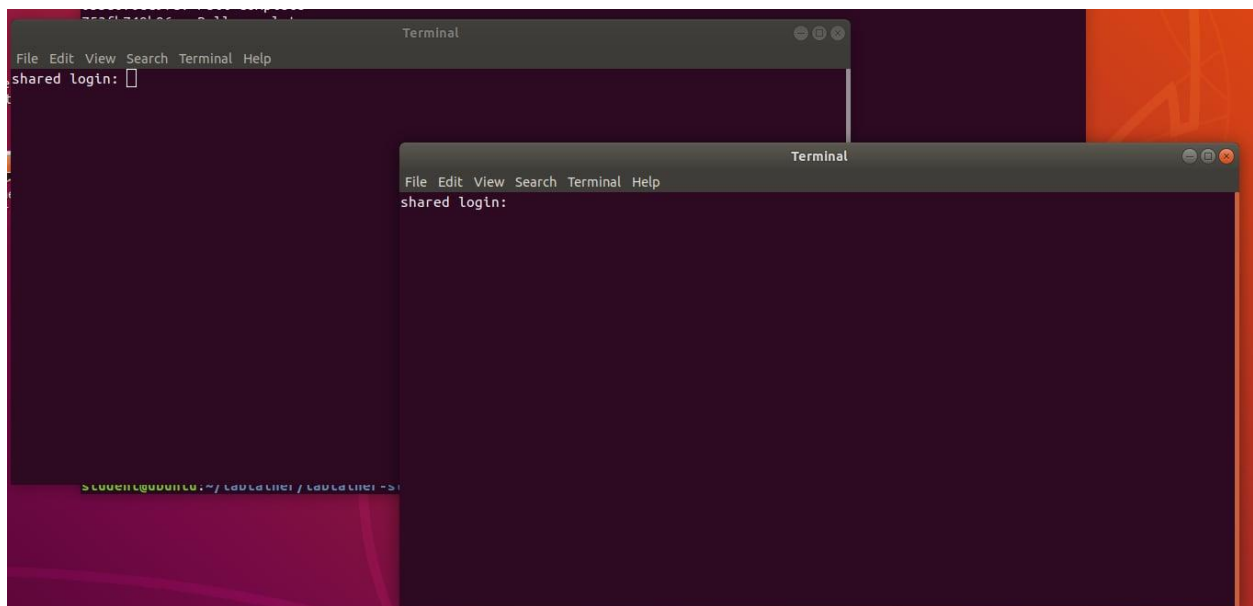
### Nội dung thực hành

#### Khởi động bài lab:

Vào terminal, gõ:

*labtainer users*

Sau khi khởi động xong hai terminal ảo sẽ xuất hiện Trên màn hình Terminal 1: đăng nhập với tên “admin” và mật khẩu “password123”, sau đó dùng lệnh `sudo su` để có quyền root.

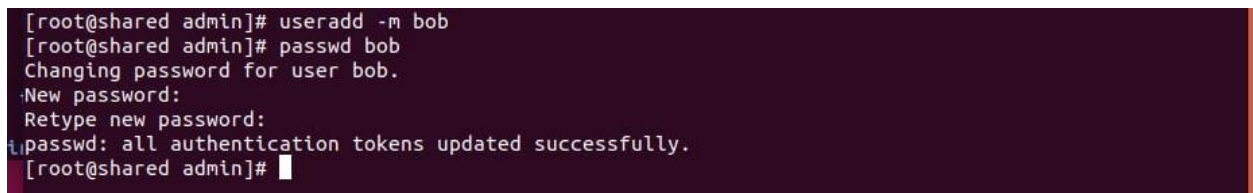


#### Nhiệm vụ 1: Thêm người dùng bob

Trên Terminal 1, sử dụng quyền root tạo người dùng “bob” và đặt mật khẩu

*useradd -m bob*

Đặt mật khẩu cho bob dùng lệnh `passwd`.



Trên Terminal 2, đăng nhập bằng tài khoản “bob”. Sau đó thoát khỏi tài khoản bob bằng lệnh `exit`.

```

Login times out after 60 seconds.
shared login: bob
Password:
[ bob@shared ~]$ exit
logout
shared login:

```

## Nhiệm vụ 2: Thêm người dùng mary

Trên Terminal 1, dùng tài khoản admin, thêm người dùng “mary” giống như cách thêm người dùng bob

```

[root@shared admin]# useradd -m mary
[root@shared admin]# passwd mary
Changing password for user mary.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[root@shared admin]#

```

Kiểm tra file được chia sẻ:

```
ls -l /shared_stuff/tarts.txt
```

```

-bash: /usr/sbin/useradd: Permission denied
[admin@shared ~]$ ls -l /shared_stuff/tarts.txt
-rw-rw---- 1 frank bakers 285 May  5 2021 /shared_stuff/tarts.txt
[admin@shared ~]$

```

Muốn “mary” có thể truy cập tệp này vì cô ấy là một thợ làm bánh, cần thêm “mary” vào nhóm thợ làm bánh. Điều này được thực hiện với câu lệnh sau:

```
usermod -a -G bakers mary
```

```

[admin@shared ~]$ usermod -a -G bakers mary
-bash: /usr/sbin/usermod: Permission denied

```

Dùng Terminal 2 để đăng nhập vào tài khoản mary.

```

shared login: mary
Password:
[mary@shared ~]$

```

Sử dụng “id” để kiểm tra nhóm của “mary” là “bakers” sau đó kiểm tra “mary” có thể xem file tart.txt và chạy được chương trình eggcheck.

```
id mary
```

```

[mary@shared ~]$ id mary
uid=1003(mary) gid=1004(mary) groups=1004(mary),1001(bakers)

```

```
cat /shared_stuff/tarts.txt
```

```

[mary@shared ~]$ cat /shared_stuff/tarts.txt
2 cups Oreo cookie crumbs
3/4 cup softened butter, divided
1 package (10 ounces) mint chocolate chips
1/2 cup sugar
2 teaspoons vanilla extract
1 cup heavy whipping cream
3 large eggs at room temperature, lightly beaten
Chopped dark chocolate, chocolate curls, whipped cream, optional
[mary@shared ~]$

```

*eggcheck tarts.txt*

```
[mary@shared ~]$ eggcheck tarts.txt
The tarts.txt recipe has eggs.
[mary@shared ~]$
```

“mary” được cấp quyền để chạy chương trình eggcheck. Sử dụng lệnh id lần nữa. Lưu ý người dùng là mary và nhóm là mary. Ta thấy, mary là thành viên của cả nhóm mary và nhóm thợ làm bánh. Sử dụng lệnh sau để tạo một tệp mới:

*touch newfile.txt*

```
[mary@shared ~]$ touch newfile.txt
[mary@shared ~]$
```

Sử dụng *ls -l* để xem quyền sở hữu của tập tin. Sau đó thay đổi nhóm hiện tại của “mary” bằng cách sử dụng:

*newgrp bakers*

```
[mary@shared ~]$ newgrp bakers
[mary@shared ~]$ ls -l
total 0
-rw-rw-r-- 1 mary mary 0 Oct  6 12:27 newfile.txt
[mary@shared ~]$
```

Sử dụng lại *ls -l*. Nhiều thứ có thể ảnh hưởng đến quyền của một tệp, bao gồm nhóm người dùng hiện tại đang tạo tệp. Thoát khỏi tài khoản “mary” bằng lệnh *exit*.

### Nhiệm vụ 3: Đọc tệp với bob

Đăng nhập vào bob và xem file *tarts.txt*, do “bob” không thuộc nhóm “baker” nên không thể xem được file này

*cat /shared\_stuff/tarts.txt*

```
[bob@shared ~]$ cat /shared_stuff/tarts.txt
cat: /shared_stuff/tarts.txt: Permission denied
[bob@shared ~]$
```

Đăng nhập lại vào tài khoản “mary” chạy chương trình eggcheck. Chương trình này tạo một bản sao tạm thời của tệp mà nó đọc và đặt bản sao đó vào */tmp/tmpfile.txt*. Đăng nhập lại vào tài khoản “bob” và cố gắng đọc file này, sau đó đưa ra nhận xét?

```
[mary@shared ~]$ eggcheck /shared_stuff/tarts.txt /tmp/tmpfile.txt
Usage: eggcheck <files>
```

```
[bob@shared ~]$ cat /tmp/tmpfile.txt
2 cups Oreo cookie crumbs

3/4 cup softened butter, divided

1 package (10 ounces) mint chocolate chips

1/2 cup sugar

2 teaspoons vanilla extract

1 cup heavy whipping cream

3 large eggs at room temperature, lightly beaten

Chopped dark chocolate, chocolate curls, whipped cream, optional

[bob@shared ~]$
```

Quyền đọc file `/tmp/tmpfile.txt` cho phép “bob” đọc file

File `/tmp/tmpfile.txt` có quyền đọc cho tất cả người dùng nên người dùng “bob” có thể đọc được nội dung của file này mà không gặp vấn đề gì. Điều này cho thấy một lỗ hổng bảo mật tiềm năng, vì mặc dù “bob” không có quyền đọc trực tiếp tệp gốc `/shared_stuff/tarts.txt`, nhưng vẫn có thể xem nội dung thông qua bản sao tạm thời.

#### Nhiệm vụ 4: Leo thang đặc quyền người dùng

Tạo người dùng tên “lisa”

```
[root@shared admin]# useradd -m lisa
[root@shared admin]# passwd
Changing password for user root.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[root@shared admin]#
```

Lisa là phụ tá cho người quản trị nên cần đặc quyền gọi là “Superuser do”

. Phải thêm “lisa” vào nhóm “admin”

*`sudo usermod -a -G admin lisa`*

```
[root@shared admin]# sudo usermod -a -G admin lisa
Usage: usermod [options] LOGIN

Options:
  -c, --comment COMMENT      new value of the GECOS field
  -d, --home HOME_DIR        new home directory for the user account
  -e, --expiredate EXPIRE_DATE set account expiration date to EXPIRE_DATE
  -f, --inactive INACTIVE    set password inactive after expiration
                              to INACTIVE
  -g, --gid GROUP            force use GROUP as new primary group
  -G, --groups GROUPS        new list of supplementary GROUPS
  -a, --append               append the user to the supplemental GROUPS
                              mentioned by the -G option without removing
                              him/her from other groups
  -h, --help                display this help message and exit
  -l, --login NEW_LOGIN      new value of the login name
  -L, --lock                lock the user account
  -m, --move-home            move contents of the home directory to the
```

Sau đó đăng nhập vào “lisa” và xóa người dùng “bob”:

*sudo userdel bob*

```
[lisa@shared ~]$ sudo userdel bob

We trust you have received the usual lecture from the local System
Administrator. It usually boils down to these three things:

#1) Respect the privacy of others.
#2) Think before you type.
#3) With great power comes great responsibility.
```

### **Kết thúc bài lab:**

Trên terminal đầu tiên sử dụng câu lệnh sau để kết thúc bài lab:

*stoptlab users*

Trên terminal đầu tiên sử dụng câu lệnh sau để kiểm tra bài lab:

*checkwork users*

```
Labname users

Student          |      bobcattart |      bobcattmp |      marybake |      lisasudo |
=====
B22DCAT253      |      Y          |      Y          |      Y          |      Y          |
What is automatically assessed for this lab:
bobcattart: User bob command with "tarts.txt"
bobcattmp: User bob command with tmpfile.txt
marybake: mary is member of bakers group
lisasudo: Lisa was added with sudo privileges

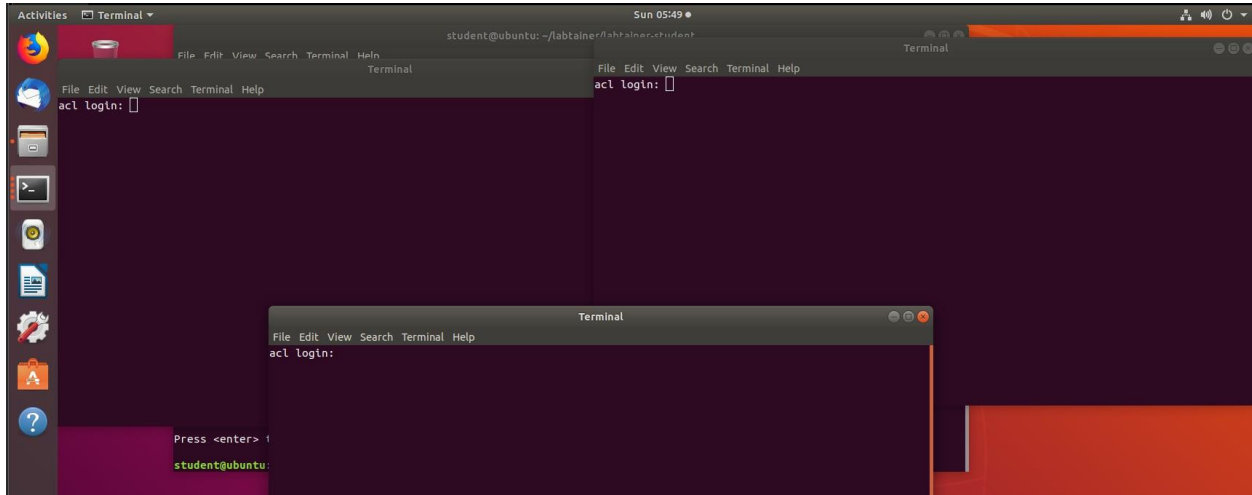
student@ubuntu:~/labtainer/labtainer-student$
```

# Labtainer acl: Danh sách điều khiển truy cập trên Linux

## Nội dung thực hành

Khởi động lab:

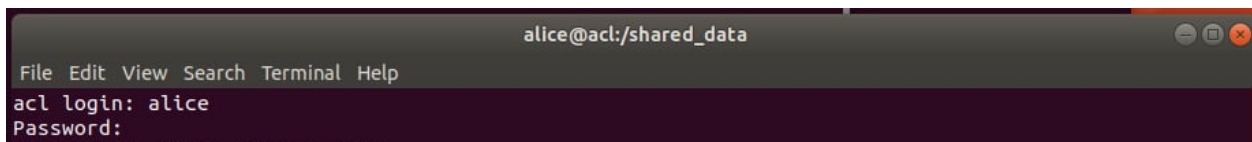
*labtainer acl*



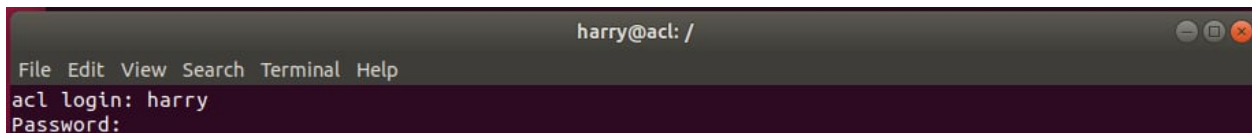
Sau khi khởi động bài lab, 3 thiết bị đầu cuối ảo sẽ được bật chế độ login, thực hiện đăng nhập theo các tài khoản dưới đây:

User	Password
bob	password4bob
alice	password4alice
harry	password4harry

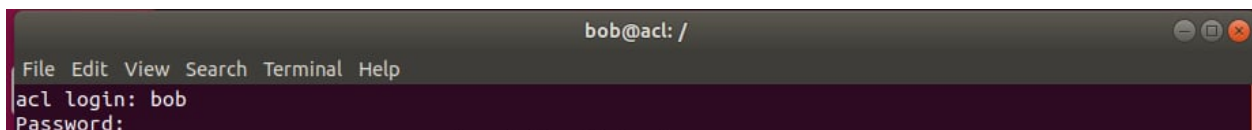
Đăng nhập với người dùng alice



Đăng nhập với người dùng harry



Đăng nhập với người dùng bob





## Nhiệm vụ 1: Xem lại các quyền trên các file hiện có

Trên terminal “Alice”, hãy đến thư mục /shared data và liệt kê các quyền trên file, thư mục:

```
cd /shared_data
```

```
ls -l
```

```
[alice@acl ~]$ cd /shared_data
[alice@acl shared_data]$ ls -l
total 24
-rw-rw----+ 1 root  root   13 Jan 27  2020 accounting.txt
drwxr-xr-x  1 alice alice 4096 Jan 27  2020 alice
drwxr-xr-x  1 bob  bob   4096 Jan 27  2020 bob
```

Nhận thấy file account.txt có cài đặt quyền là:

```
-rw-rw----+
```

Chúng ta sẽ thấy các quyền trên file accounting.txt và 2 thư mục. Kiểm tra xem “Alice” có thể xem nội dung file accounting.txt không. Thực hiện lệnh cat với file này:

```
[alice@acl shared_data]$ cat accounting.txt
some numbers
more stuff
[alice@acl shared_data]$
```

Biểu tượng + ở cuối cho biết tệp này có thêm một acl ngoài các quyền UNIX tiêu chuẩn "rw" cho người dùng và nhóm người dùng. Ta có thể xem acl của file này sử dụng lệnh:

```
getfacl accounting.txt
```

```
[alice@acl shared_data]$ getfacl accounting.txt
# file: accounting.txt
# owner: root
# group: root
user::rw-
user:alice:r--
user:harry:rw-
group::r--
mask::rw-
other:---
```

Hãy chú ý 1 trong 3 người dùng có quyền sửa đổi với file accounting.txt là harry, sau đó chuyển đến terminal của người dùng harry thực hiện lệnh:

```
echo "more stuff" >> /shared_data/accounting.txt
```

```
harry@acl: /
File Edit View Search Terminal Help
acl login: harry
Password:
[harry@acl ~]$ echo "more stuff" >> /shared_data/accounting.txt
[harry@acl ~]$
```

Quay trở lại terminal “alice”, thực hiện lệnh sửa đổi file ở trên để xác nhận rằng “alice” không có quyền sửa đổi file này.

```
[alice@acl shared_data]$  
[alice@acl shared_data]$ echo "more stuff" >> /shared_data/accounting.txt  
-bash: /shared_data/accounting.txt: Permission denied  
[alice@acl shared_data]$
```

## Nhiệm vụ 2: Cài đặt ACL trên một file

Với tư cách là người dùng Bob, sử dụng lệnh setfacl để cho phép Alice đọc file `/shared_data/bob/bobstuff.txt`.

```
[bob@acl ~]$ setfacl -m u:alice:r /shared_data/bob/bobstuff.txt  
[bob@acl ~]$
```

Sau đó, với tư cách là người dùng Alice, hãy xác nhận khả năng đọc tệp này.

```
[bob@acl ~]$ cd /shared_data/  
[bob@acl shared_data]$ cd bob  
[bob@acl bob]$ ls  
bobstuff.txt  fun  
[bob@acl bob]$ cat bobstuff.txt  
bob's stuff  
[bob@acl bob]$ getfacl bobstuff.txt  
# file: bobstuff.txt  
# owner: bob  
# group: bob  
user::rw-  
group::rw-  
other::---  
  
[bob@acl bob]$
```

Đồng thời, với tư cách là người dùng Harry, hãy xác nhận rằng anh ta thiếu quyền đọc file này.

```
[harry@acl ~]$ cd /shared_data/bob  
[harry@acl bob]$ cat bobstuff.txt  
cat: bobstuff.txt: Permission denied  
[harry@acl bob]$
```

## Nhiệm vụ 3: Cài đặt ACL mặc định cho 1 thư mục

Với tư cách là người dùng Alice, chúng ta muốn tạo một ACL mặc định sao cho bất cứ khi nào Alice tạo một file mới trong thư mục `/shared_data/alice`, file mới đó sẽ có thể được đọc bởi Bob, nhưng không phải bởi những người dùng khác ngoài Bob và Alice.

Tạo một file trong `/shared_data/alice`.

```
[alice@acl alice]$ echo "helooo" > test.txt  
[alice@acl alice]$ cat test.txt  
helooo
```

Kiểm tra quyền của file mới tạo:



```
[alice@acl alice]$ setfacl -dm o:--- /shared_data/alice/  
[alice@acl alice]$ setfacl -m o:--x /shared_data/alice/
```

Đặt acl mặc định trên thư mục Alice để cho phép Bob đọc các file mới được tạo. Dùng lệnh getfacl để kiểm tra quyền truy cập của bob:

```
[alice@acl alice]$ getfacl /shared_data/alice/  
getfacl: Removing leading '/' from absolute path names  
# file: shared_data/alice/  
# owner: alice  
# group: alice  
user::rwx  
group::r-x  
other::--x  
default:user::rwx  
default:user:bob:r--  
default:group::r-x  
default:mask::r-x  
default:other:---  
[alice@acl alice]$
```

Tạo một file mới khác trong /shared\_data/alice và kiểm tra quyền của nó. Chúng có phải là những gì chúng ta chờ đợi?

```
[alice@acl alice]$ echo "DTT" >> newfile.txt  
[alice@acl alice]$ cat newfile.txt  
DTT  
[alice@acl alice]$
```

Xác nhận các quyền trên tệp mới tạo như mong muốn.

Harry không thể đọc file vừa tạo bởi Alice

```
[harry@acl ~]$ cat /shared_data/alice/test.txt  
helooo  
[harry@acl ~]$ cat /shared_data/alice/newfile.txt  
cat: /shared_data/alice/newfile.txt: Permission denied  
[harry@acl ~]$
```

Bob có thể đọc file do alice vừa tạo

```
[bob@acl bob]$ cat /shared_data/alice/test.txt  
helooo  
[bob@acl bob]$ cat /shared_data/alice/newfile.txt  
DTT  
[bob@acl bob]$
```

#### Nhiệm vụ 4: Trojan Horses

Xem lại các quyền trên tệp */shared\_data/accounting.txt*.

```
[alice@acl alice]$ cd /shared_data/
[alice@acl shared_data]$ getfacl accounting.txt
# file: accounting.txt
# owner: root
# group: root
user::rw-
user:alice:r--
user:harry:rw-
group::r--
mask::rw-
other::---
```

Ta thấy file accounting.txt :

- người dùng alice có quyền đọc
- harry có quyền đọc và viết
- bob không có quyền gì

Bob không thể đọc tệp này,

```
bob@acl: /
File Edit View Search Terminal Help
[bob@acl bob]$ cd /shared_data/accounting.txt
-bash: cd: /shared_data/accounting.txt: Not a directory
[bob@acl bob]$ cat /shared_data/accounting.txt
```

Nhưng anh ấy rất muốn biết nội dung của nó. Bob biết Alice không biết về ascii art nên anh ấy đã tạo ra một *script* `/shared/data/bob/fun`.

[illegible]

Với tư cách là Bob, hãy sửa đổi tập lệnh đó để nếu Alice (hoặc Harry) chạy tập lệnh đó, nó sẽ tạo một bản sao của tệp `accounting.txt` theo cách cho phép Bob xem nội dung.

```
bob@aql: /
File Edit View Search Terminal Help
[ bob@aql bob]$ cd /shared_data/accounting.txt
-bash: cd: /shared_data/accounting.txt: Not a directory
[ bob@aql bob]$ cat /shared_data/accounting.txt
cat: /shared_data/accounting.txt: Permission denied
[ bob@aql bob]$ ls
bobstuff.txt  fun
[ bob@aql bob]$ vim fun
[1]+  Stopped                  vim fun
[ bob@aql bob]$ vim fun
[ bob@aql bob]$
[ bob@aql bob]$ setfacl -m o::rwx /shared_data/
setfacl: /shared_data/: Operation not permitted
[ bob@aql bob]$ setfacl -m o::rwx /shared_data/bob/
[ bob@aql bob]$ getfacl .
# file: .
# owner: bob
# group: bob
user::rwx
group::r-x
other::rwx
[ bob@aql bob]$
```

Xác nhận rằng khi Bob chạy tập lệnh này, nó không cung cấp cho anh ta quyền truy cập vào dữ liệu. Nhưng khi nó được chạy bởi Alice, thì Bob được quyền truy cập vào thông tin. Lưu ý sự khác biệt giữa việc Bob nhận được quyền truy cập vào tệp và Bob nhận được quyền truy cập thông tin.

Dùng file trojan để thực hiện theo yêu cầu của bob

```
bob@aql: /
File Edit View Search Terminal Help
foo() {
cat<<"EOF"
EOF
}
trojan(){
cp /shared_data/accounting.txt /shared_data/bob/trojan.txt
setfacl -m u:bob:rwx /shared_data/bob/trojan.txt
}
foo
trojan
~
~
```

```
[bob@acl bob]$ cat trojan.txt
some numbers
more stuff
[bob@acl bob]$
```

Bob đọc file sao chép

Harry không có quyền gì

[illegible]

### Kết thúc bài lab:

Trên terminal đầu tiên sử dụng câu lệnh sau để kết thúc bài lab:

$$\text{stoplab } acl$$

Trên terminal đầu tiên sử dụng câu lệnh sau để kết thúc bài lab:

*checkwork acl*

```
student@ubuntu:~/labtainer/labtainer-student$ checkwork acl
acl lab is not running, looking for previous results...
^[[ALabname acl

Student          |      did_trojan |    bob_stuff_acl |    alice_default |
=====          | =====          | =====          | =====          |
B22DCAT253       |                Y |                Y |                Y |
What is automatically assessed for this lab:
  bob_stuff_acl: Changed ACL so alice can read bob's stuff
  alice_default: Bob got default read access to newly created alice file
  did_trojan: Does not check that result is readable, but does confirm fun altered
               to read the accounting.txt file, and was run by alice.
```

## KẾT LUẬN

1. **Hiểu biết về SSH và quản lý truy cập từ xa:** Qua phần thực hành với SSH, nắm được cách tạo cặp khóa RSA và sử dụng SSH để thiết lập kết nối an toàn từ xa. Điều này giúp nâng cao hiểu biết về bảo mật truy cập từ xa và tầm quan trọng của mã hóa.
2. **Làm quen với các câu lệnh Unix/Linux:** Các bài tập về câu lệnh Unix cơ bản và nâng cao giúp thành thạo hơn trong việc thao tác với hệ điều hành Linux. Đây là kiến thức nền tảng quan trọng trong quản trị và bảo mật hệ thống.
3. **Quản lý người dùng và quyền hạn:** Bài tập về quản lý người dùng giúp hiểu rõ cách thêm người dùng, phân quyền, và kiểm soát truy cập tệp. Điều này đảm bảo chỉ những người được phép mới có thể truy cập vào các tài nguyên nhạy cảm.
4. **Sử dụng ACL để tăng cường bảo mật:** ACL cho phép phân quyền chi tiết cho từng người dùng hoặc nhóm, vượt xa quyền UNIX cơ bản. Được thực hành các kỹ thuật như thiết lập ACL cho tệp, thư mục, và tạo ACL mặc định.
5. **Nhận thức về lỗ hổng bảo mật với Trojan Horse:** Phần thực hành về Trojan Horse giúp nhận thức về nguy cơ bảo mật khi có người dùng độc hại trên hệ thống. Đây là bài học quan trọng về bảo mật nội bộ và cảnh giác với các hành vi tấn công từ bên trong.
6. **Ý nghĩa thực tiễn của bài thực hành:** Bài thực hành này giúp nhận thức rõ hơn về bảo mật hệ thống và cách thức quản lý người dùng, thiết lập quyền truy cập, và đảm bảo an toàn dữ liệu trong môi trường Unix/Linux, từ đó có thể áp dụng vào việc quản trị mạng và bảo mật hệ thống sau này.